



## Which Distributed Development Concept Did You Have in Mind?



Are we developing software at distributed sites, or are we developing distributed software? These are obviously very different concepts. Until I saw the articles that are in this month's issue, I didn't realize there was so much about distributed software that I didn't know.

**Distributed Development of Software:** When we received Paul McMahon's article, *Distributed Development: Insights, Challenges, and Solutions*, it occurred to me that I wish I had known this information long ago. McMahon's ideas about seamless vision, virtual communication, and testing the virtual organization would have really been helpful to one of my past projects. My company worked with another company, trying to provide the end product to the customer; it was difficult at best. This was not a joint collaboration from two companies working as a team. The customer selected my company to develop most of the software, then informed us that we would let company B develop a portion of the software and use their product with ours.

There was no attempt to integrate product teams, create a shared vision, or communicate virtually. Company B simply sent numerous versions of their software, and each time I sent them back a list of errors I found while testing it. When the software was returned, some errors were corrected, some were not, and new errors were inserted. Finally, I just sent them a copy of the test I was running so they could verify that the software passed the test before sending it to me again.

However, I did not realize that company B did not use the same computer set-up as I did. My test listed keystrokes – not computer functionality. So they sent the software back with errors in it again because they could not run my test. The other company did not know what I was trying to accomplish with the test, and instead of calling and asking for clarifications, they just sent back another bug-filled software package with an unusable user's manual. Management finally just requested that I fix the software myself and rewrite company B's user's manual. That finally brought closure, but time and effort would have been saved using McMahon's insights.

**Development of Distributed Software:** In *Toward Adaptive and Reflective Middleware for Network-Centric Combat Systems*, Dr. Douglas Schmidt and several of his colleagues propose a vision for how coupling military sponsored research, industrial standards development, and commercial products can dramatically advance capabilities in the development of complex military embedded systems. They cite the Boeing Bold Stroke project as just one example of a real-world military system leveraging this technology toward the establishment of an open systems-based software product line. We will learn more about Bold Stroke in December with an article that describes their Air Force Research Laboratory-sponsored work in affordably upgrading legacy applications to COTS-based computing platforms.

**Distributed Development of Distributed Software:** What if you want to purchase distributed software middleware? Dr. Thomas Croak comes to the rescue with *Factors to Consider When Selecting CORBA Implementations*. If you are planning to buy a CORBA distributed object-computing system this article will help. Dr. Croak's analogy to buying a car is also applicable to any software purchases you may plan to make.

Dr. Lon Gowen's supporting article, *Predicting Staff Sizes to Maintain Networks*, provides helpful information to readers responsible for managing technical resources. Predicting staffing sizes is a problem for many organizations, and this article should help with the struggle.

Hillel Glazer rounds out this issue with *Dispelling the Process Myth: Having a Process Doesn't Mean Sacrificing Agility or Creativity*. Having worked on several small projects, I agree that tailored processes can indeed make agile processes more productive with the time saved by reducing rework.

The theme topics addressed in this month's CROSSTALK are not my areas of expertise. However, the articles are very informative, and I now know more than I did a few months ago. Maybe if these subjects come up in the future, I will now know enough to start asking the right questions to help with new endeavors. I learned a lot from reading this month's CROSSTALK. Of course, I never miss an issue. (As the associate publisher, I'm paid not to miss an issue.) I hope you never miss an issue either.

Elizabeth Starrett  
Associate Publisher